



زانکۆی پۆلیتەکنیکی هەولێر

ERBIL POLYTECHNIC UNIVERSITY

(Laravel Authentication Tool Report)

Department of Information System Engineering
7th semester Information System Architecture

Prepared by:

Rezhyar Sardar

Mustafa Raad

Abdulaziz Ibrahim

Taman Sardar

Supervisor:

Mr.Hezha Ahmed

Contents

Abstract	3
Introduction	3
Project Overview.....	4
Where Authentication Is Used in the Project	5
1. Login Page.....	5
2. Login Processing.....	5
3. Protected Dashboard	5
4. Route Protection.....	5
5. Login Logging System	5
Routes and How They Use the Auth Tool.....	6
How the AuthController Uses Authentication.....	7
1-login() — Where Authentication Happens	7
2-logout() — Ending Authentication	7
How Sessions Are Used for Authentication.....	8
Where Login Logging Happens.....	8
Blade Views and Their Role in Authentication	9
Security Practices Used in This Project.....	9
Conclusion	10

Abstract

This report explains the authentication tool used in the Laravel project. The system allows users to log in, stay authenticated using sessions, access protected pages, and log out securely. It also includes a login logging feature that records every login attempt. The report focuses on **where authentication is implemented in the project and how each part of the code works** to provide secure user access control.

Introduction

The project uses Laravel's authentication system to let users log in with their credentials, access a protected dashboard, and stay logged in through session storage. It also allows users to log out safely while keeping the system secure. In addition, the project records every login attempt in the database for tracking and security purposes. Overall, the authentication tool ensures that only authorized and authenticated users can access specific parts of the application.

Project Overview

This report covers the following points:

- What the authentication tool does in this project
- Where authentication is used in the project
- How the login process works in your code
- How session-based authentication protects the dashboard
- How logout works
- Where the project logs login attempts
- Which files are responsible for authentication
- How the project applies Laravel's built-in auth features

Where Authentication Is Used in the Project

The authentication tool is used in **three main places** in your project:

1. Login Page

Located in:

`resources/views/login.blade.php`

This is where the user enters their email and password.

2. Login Processing

Handled in:

`app/Http/Controllers/AuthController.php`

This part checks the user credentials, starts the session, and logs the login attempt.

3. Protected Dashboard

Located in:

`resources/views/dashboard.blade.php`

This page is only shown if the auth tool confirms the user is logged in.

4. Route Protection

Defined in:

`routes/web.php`

Routes are protected using Laravel's `auth` middleware.

5. Login Logging System

Model:

`app/Models/LoginLog.php`

Migration:

`database/migrations/...create_login_logs_table.php`

Every login attempt is stored in the database using this tool

Routes and How They Use the Auth Tool

Login Routes:

```
Route::get('/login', [AuthController::class, 'showLogin'])->name('login');  
Route::post('/login', [AuthController::class, 'login']);
```

These routes show and process the login form.

Protected Routes Using the Auth Tool:

```
Route::middleware('auth')->group(function () {  
    Route::get('/dashboard', function () {  
        return view('dashboard');  
    });  
    Route::post('/logout', [AuthController::class, 'logout']);  
});
```

- Only authenticated users can open /dashboard.
- This is enforced by the **auth middleware**.
- The middleware checks Laravel's session to confirm the user is logged in.

This is the main place where the **auth tool is directly used**.

How the AuthController Uses Authentication

inside `AuthController.php`, the auth tool is used in two main methods:

1-login() — Where Authentication Happens

This method:

- Validates the email and password
- Attempts login using Laravel's `Auth::attempt()`
- Starts a secure session
- Saves the login attempt using `LoginLog`
- Redirects the user to the dashboard

This is the core of your authentication tool.

2-logout() — Ending Authentication

This method:

- Calls `Auth::logout()` to remove the user from the session
- Invalidates the old session
- Regenerates a token for security
- Sends the user back to the login page

This completes the authentication cycle.

How Sessions Are Used for Authentication

After Auth::attempt() succeeds:

- Laravel creates a **session**
- The auth middleware checks this session for protected pages
- If there is no session → redirect to login
- If session exists → allow access to dashboard

This session system **is the actual authentication tool inside the project.**

Where Login Logging Happens

Model:

app/Models/LoginLog.php

Migration:

create_login_logs_table.php

This records:

- User ID (if the login was successful)
- IP address
- Status (success/failed)
- Timestamp

This is part of the authentication system because it tracks all access attempts.

Blade Views and Their Role in Authentication

login.blade.php

- Shows the login form
- Uses @csrf for security
- Sends credentials to /login route

dashboard.blade.php

- Only shown when auth tool confirms session is valid
- Includes logout button

The dashboard itself demonstrates **where the auth tool controls access**.

Security Practices Used in This Project

- CSRF protection
- Session regeneration after login
- Session invalidation on logout
- Middleware protection of routes
- Logging of login attempts

Conclusion

In conclusion, this project uses Laravel's **authentication** tool to manage user login, keep users logged in with **sessions**, protect certain pages, record login attempts, and handle logout safely. The login page, the login process, the protected dashboard, and the logging system all show how the auth tool works in this project. Overall, the setup is simple, clear, and does a good job of controlling who can access the system.